

Implementace interpretu

Interpret je implementován v 6ti souborech *interpret.py*, *argsparse.py*, *xmlparse.py*, *instructions.py*, *runtime.py* a *errors.py*.

Argumenty interpretu

Usage: interpreter.py [-help] [-source=SOURCE] [-input=INPUT]

Na začátku interpretu se kontrolují argumenty vytvořením instance třídy *Argparse* a zavoláním metody *parse()*, která argumenty zkontroluje a přiřadí hodnoty daným proměnným.

Metoda *parse* nejdříve zkontroluje, jestli nebyl jeden z argumentů *-help*. Pokud ano, podívá se, jestli byl interpret spouštěn s jakýmkoliv jiný argumentem, což by znamenalo chybu. Je-li interpret spouštěn pouze s *help* parametrem, vypíše *help* a exitne.

Pokud se v argumentech *help* nenachází, podívá se jestli je interpret volán aspon jedním z *-source* a *-input* parametrů. Pokud oba chybí, vede to k chybě. Chybí-li pouze jeden, je chybějící soubor načítán ze standartního inputu.

Validace XML kódu

Nejdříve se vytvoří instance třídy *XmlParser* a zavoláním metody *parse(file)* začne samotná validace XML reprezentace kódu.

Na validaci XML používám knihovnu *xml.etree.ElementTree*. Nejdříve je do funkce z knihovny *etree* *fromstring* vložen obsah source souboru nebo obsah *stdin* (záleží zda byl soubor soubor přidán jako argument). Dále je zavolána metoda *validate_header*, která zkontroluje hlavičku XML. Pokud kontrola v pořádku proběhne, zavolá se metoda *check_inst(root)*, která projde všechny instrukce, zda mají správné pořadí a příkaz (opcode). Dále se informace o instrukci vloží do listu instrukce a pomocí metody *check_args(instruction, attrs, arg_num)* se zkontrolují argumenty instrukce. Kontroluje se jejich počet a zda jejich hodnota odpovídá jejich typu. Jejich informace jsou podle pořadí argumentů vloženy do listu instrukce a takto zkontrolovaná instrukce je vložena do listu programu, který je poté podle pořadí seřazen a samotné pořadí je z každého listu instrukce odstraněno.

Postupné provádění kódu

Začíná vytvořením instance *run* třídy *Instructions* a zavoláním jediné metody třídy *handle_instructions(xml.program)*, ve které se vytvoří instance třídy *Runtime*.

Nejdříve projde interpret celým listem instrukcí a všechny labely, které nalezne vloží do listu labelů. Poté projde všemi instrukcemi znovu a podle názvu instrukce zavolá příslušnou metodu třídy *Runtime*. Podle toho o jakou instrukci se jedná se může nějaká instrukce přeskočit nebo zopakovat. Jakmile takto dojdeme na konec listu instrukcí, znamená to konec interpretu.

Rozšíření

Rozšíření není z časových důvodů implementované žádné. Myslím si že například pro float by stačilo kód upravit bez potřeby upravovat návrh.

Class diagram

